

## **DETAILED ACTION**

1. The reply filed July 29, 2008, has been received and entered. Claims 1, 4, 5, 8-12, 15, and 18-40 are pending.

### ***Response to Arguments***

2. Applicant's arguments filed July 29, 2008, have been fully considered but they are not persuasive.

Although, Pleso (in col. 1, lines 43-54) distinguishes a device driver from a specific separate application, Pleso does not suggest that device drivers themselves are not applications. To the contrary, Pleso discloses that the device drivers may be JAVA applications (see col. 13, line 36, through col. 14, line 26). A program created using the JAVA programming language is JAVA application program. Therefore, applicant's argument that Pleso's disclosure of downloading a device driver does not include downloading an application (Remarks p. 3) is unpersuasive. Further, determining whether a device driver is needed meets the limitation of determining if the host device comprises a copy of the software program (the JAVA application device driver of Pleso).

### ***Claim Rejections - 35 USC § 102***

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4. Claims 1, 4, 5, 8-12, 15, 21-23, and 25 are rejected under 35 U.S.C. 102(b) as being anticipated by U.S. Patent No. 6,009,480 (Pleso).

As per claim 1, *Pleso* discloses: A method of installing a software program in a host device (see, e.g., col. 3, lines 18-31 (method for installing a peripheral device driver to a computer system)), said software program being required for said host device to communicate with a peripheral device (see, e.g., Abstract (a device driver which the computing system employs to communicate with the peripheral system)), said method comprising the steps of:

coupling said host device to said peripheral device (see, e.g., col. 3, lines 18-31 (connecting a peripheral device to a computer)) utilizing a USB serial interface, said peripheral device containing said software program stored in a memory device contained in said peripheral device (see, e.g., col. 3, lines 18-31 (the peripheral device including . . . a first memory . . . storing the peripheral device driver));

uploading said software program from said peripheral device to said host device over said USB serial interface (see, e.g., col. 3, lines 5-16 (the peripheral device downloads the device driver to the second memory . . . ); col. 11, lines 55-65 (USB)); and

installing said software program in said host device (see, e.g., col. 3, lines 5-16 (the peripheral device downloads the device driver to the second memory . . . )) thereby allowing communication between said host device and said peripheral device (see, e.g., Abstract (a device driver which the computing system employs to communicate with the peripheral system));

wherein said software program is an application for executing a functional operation associated with the operation of the peripheral device (see, e.g., col. 3, lines 18-31 (method for installing a peripheral device driver to a computer system); col. 13, line 36, through col. 14, line 26 (the device drivers may be JAVA applications)).

As per claim 4, *Pleso* further discloses said host device and said peripheral device communicate with one another utilizing the USB specification (see, e.g., col. 11, lines 55-65).

As per claim 5, *Pleso* discloses: A method of installing a software program in a host device (see, e.g., col. 3, lines 18-31 (method for installing a peripheral device driver to a computer system)), said software program being required for said host device to communicate with a peripheral device (see, e.g., Abstract (a device driver which the computing system employs to communicate with the peripheral system)), said method comprising the steps of:

coupling said host device to said peripheral device (see, e.g., col. 3, lines 18-31 (connecting a peripheral device to a computer)) utilizing a USB serial interface, said peripheral device containing said software program stored in a memory device contained in said peripheral device (see, e.g., col. 3, lines 18-31 (the peripheral device including . . . a first memory . . . storing the peripheral device driver));

[determining] if said host device comprises said software program in said memory device contained in said host device (see, e.g., Figure 8 (step 252); col. 11, lines 7-10 (determines whether a peripheral device driver is needed)) [and if not,] uploading said software program from said peripheral device to said host device over said USB serial interface (see, e.g., col. 3, lines 5-16 (the peripheral device downloads the device driver to the second memory . . . ); col. 11, lines 55-65 (USB));

installing said software program in said host device (see, e.g., col. 3, lines 5-16 (the peripheral device downloads the device driver to the second memory . . . )) thereby allowing communication between said host device and said peripheral device (see, e.g., Abstract (a device driver which the computing system employs to communicate with the peripheral system));

wherein said software program is an application for executing a functional operation associated with the operation of the peripheral device (see, e.g., col. 3, lines 18-31 (method for installing a peripheral device driver to a computer system); col. 13, line 36, through col. 14, line 26 (the device drivers may be JAVA applications).

As per claim 8, *Pleso* further discloses said host device and said peripheral device communicate with one another utilizing the USB specification (see, e.g., col. 11, lines 55-65).

As per claim 9, *Pleso* discloses: A host device capable of communicating with any one of a plurality of peripheral devices utilizing a USB serial interface (see, e.g., col. 3, lines 18-31 (method for installing a peripheral device driver to a computer system); col. 11, lines 55-65 (USB)), said host device comprising:

a USB interface capable of defining said host device as a master device relative to said plurality of peripheral devices, (see, e.g., col. 3, lines 18-31 (connecting a peripheral device to a computer); col. 11, lines 55-65 (USB));

an application downloader for downloading the software driver of a given one of said plurality of peripheral devices, which is coupled to said host device via said USB serial interface (see, e.g., col. 3, lines 5-16 (the peripheral device downloads the device driver to the second memory . . . )); col. 11, lines 55-65 (USB)), and

an application installer for installing said software driver downloaded from said given one of said plurality of peripheral devices (see, e.g., col. 3, lines 5-16 (the peripheral device downloads the device driver to the second memory . . . )) so as to allow communication between said host device and said given one of said plurality of peripheral devices (see, e.g., Abstract (a

device driver which the computing system employs to communicate with the peripheral system));

wherein said application is for executing a functional operation associated with the operation of the peripheral device (see, e.g., col. 3, lines 18-31 (method for installing a peripheral device driver to a computer system); col. 13, line 36, through col. 14, line 26 (the device drivers may be JAVA applications).

As per claim 10, *Pleso* further discloses said host device and said given one of said plurality of peripheral devices communicate with one another utilizing the USB specification (see, e.g., col. 11, lines 55-65).

As per claim 11, *Pleso* discloses a peripheral device capable of communicating with a host device utilizing a USB serial interface (see, e.g., col. 3, lines 18-31 (method for installing a peripheral device driver to a computer system); col. 11, lines 55-65 (USB)), the peripheral device comprising:

a memory device (see, e.g., col. 7, lines 49-51);

a USB interface capable of defining said peripheral device as a slave device relative to said host device, (see, e.g., col. 3, lines 18-31 (connecting a peripheral device to a computer); col. 11, lines 55-65 (USB));

an application in said memory device for communicating with said host device (see, e.g., col. 7, lines 49-51); and

an application uploader for uploading the software driver to said host device (see, e.g., col. 3, lines 5-16 (the peripheral device downloads the device driver to the second memory . . . ); col. 11, lines 55-65 (USB));

wherein said application is for executing a functional operation associated with the operation of the peripheral device (see, e.g., col. 3, lines 18-31 (method for installing a peripheral device driver to a computer system); col. 13, line 36, through col. 14, line 26 (the device drivers may be JAVA applications)).

Regarding claims 12 and 15, in addition to the disclosure applied above to claims 1 and 5, the device enumeration is part of the standard USB standard discussed above. *See, e.g., Pleso* at col. 12, lines 53-66.

Regarding claim 21, *Pleso* further discloses:

a software driver downloader for downloading the software driver of a given one of said plurality of peripheral devices, which is coupled to said host device via said USB serial interface (see, e.g., col. 3, lines 5-16 (the peripheral device downloads the device driver to the second memory . . . ); col. 11, lines 55-65 (USB)), and

a software driver installer for installing said software driver downloaded from said given one of said plurality of peripheral devices (see, e.g., col. 3, lines 5-16 (the peripheral device downloads the device driver to the second memory . . . )) so as to allow communication between said host device and said given one of said plurality of peripheral devices (see, e.g., Abstract (a device driver which the computing system employs to communicate with the peripheral system));

wherein said driver is associated with said given one of said plurality of peripheral devices which allows for communication between said host device and said given one of said plurality of peripheral devices (see, e.g., Abstract (a device driver which the computing system employs to communicate with the peripheral system)).

Regarding claim 22, *Pleso* further discloses a memory device for storing said downloaded driver until said application is downloaded (see, e.g., system memory 18 in Fig. 1).

As per claim 23, *Pleso* further discloses said host device and said peripheral device communicate with one another utilizing the USB specification (see, e.g., col. 11, lines 55-65).

Regarding claim 25, *Pleso* further discloses:

a software driver in said memory device for communicating with said host device (see, e.g., col. 7, lines 49-51); and

a software driver uploader for uploading the software driver to said host device (see, e.g., col. 3, lines 5-16 (the peripheral device downloads the device driver to the second memory . . . ); col. 11, lines 55-65 (USB));

wherein said driver is associated with said given one of said plurality of peripheral devices which allows for communication between said host device and said given one of said plurality of peripheral devices (see, e.g., Abstract (a device driver which the computing system employs to communicate with the peripheral system)).

#### ***Claim Rejections - 35 USC § 103***

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. Claims 18-20 and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,009,480 (*Pleso*) and Srinivas Yarra, "USB OTG software frees dual-role handheld devices," May 16, 2002, EDN, pp. 83, 84, 86, and 88 (hereinafter "Yarra").

Regarding claims 18-20 and 24, Although *Pleso* discloses the use of a USB interface for host-peripheral communications, *Pleso* is silent on the use of the USB On-The-Go (OTG) specification to define roles of host devices and peripheral devices. However, *Yarra* teaches the OTG specification as being a supplement to the USB standard and adding several unique advantages over the previous standard, including the determination of host and peripheral roles (first through a cable select mechanism and additionally through software negotiation of roles).

*See, e.g., Yarra* at pp. 83-84 (The A device is the default host and the B device is the default peripheral. Through the Host Negotiation Protocol, the host functions can be transferred to the B device). Therefore, because the USB OTG specification, (1) was a known supplement to the standard USB specification, (2) provides known tangible benefits over the USB specification, and (3) is being used for its intended purpose (i.e., determining the role of host and peripheral in serial communications), it would have been obvious to one of ordinary skill in the art at the time the invention was made to utilize such a USB OTG specification with the system of *Pleso*.

7. Claims 26-32 and 34-39 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,009,480 (*Pleso*) and U.S. Patent No. 6,754,725 (*Wright et al.*).

Regarding claim 26, in addition to the disclosure and teachings applied above to claim 25, although *Pleso* fails to expressly disclose the application loader uploading the application after a memory device contained in the host device stores the uploaded driver, *Wright et al.* teaches that, in the context of providing drivers from a peripheral device through a USB interface to a host device, it has been known to similarly transfer other files (including program (application) files) relating to the peripheral device for separate installation on the host device or

automatic installation during the driver installation (see col. 5, lines 63-67). Therefore, it would have been obvious to one of ordinary skill in the art to incorporate such a subsequent application upload as per the driver and application installation system taught by Wright in order to gain the advantage of combining related installations for a peripheral device.

Regarding claim 27, *Pleso* discloses: A method of installing a driver/application in a host device (see, e.g., col. 3, lines 18-31 (method for installing a peripheral device driver to a computer system)), said driver/application being required for said host device to communicate with a peripheral device (see, e.g., Abstract (a device driver which the computing system employs to communicate with the peripheral system)), said method comprising the steps of:

coupling said host device to said peripheral device (see, e.g., col. 3, lines 18-31 (connecting a peripheral device to a computer)) utilizing a USB serial interface, said peripheral device containing said software program stored in a memory device contained in said peripheral device (see, e.g., col. 3, lines 18-31 (the peripheral device including . . . a first memory . . . storing the peripheral device driver));

uploading said driver from said peripheral device to said host device over said USB serial interface (see, e.g., col. 3, lines 5-16 (the peripheral device downloads the device driver to the second memory . . . ); col. 11, lines 55-65 (USB)); and

installing said driver in said host device (see, e.g., col. 3, lines 5-16 (the peripheral device downloads the device driver to the second memory . . . )) thereby allowing communication between said host device and said peripheral device (see, e.g., Abstract (a device driver which the computing system employs to communicate with the peripheral system));

wherein said driver is associated with said peripheral device which allows for communication between said host device and said peripheral device (see, e.g., col. 3, lines 18-31 (method for installing a peripheral device driver to a computer system); col. 13, line 36, through col. 14, line 26 (the device drivers may be JAVA applications).

Although in the context discussed above, *Pleso* fails to expressly disclose the uploading and installing steps being carried out for a separate driver and application (the application being for executing a functional operation associated with the operation of the peripheral device), *Wright et al.* teaches that, in the context of providing drivers from a peripheral device through a USB interface to a host device, it has been known to similarly transfer other files (including program (application) files) relating to the peripheral device (for executing a functional operation associated with the operation of the peripheral device) for separate installation on the host device or automatic installation during the driver installation (see col. 5, lines 63-67). Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate such additional application uploading and installing as per the teachings of *Wright* in order to gain the advantage of combining related installations for a peripheral device.

Regarding claim 28, in addition to the disclosure and teachings applied above to claim 25, although *Pleso* fails to expressly disclose the application loader uploading the application after a memory device contained in the host device stores the uploaded driver, *Wright et al.* teaches that, in the context of providing drivers from a peripheral device through a USB interface to a host device, it has been known to similarly transfer other files (including program (application) files) relating to the peripheral device for separate installation on the host device or automatic installation during the driver installation (see col. 5, lines 63-67). Therefore, it would

have been obvious to one of ordinary skill in the art to incorporate such a subsequent application upload as per the driver and application installation system taught by Wright in order to gain the advantage of combining related installations for a peripheral device.

Regarding claim 29, *Pleso* further discloses said host device and said peripheral device communicate with one another utilizing the USB specification (see, e.g., col. 11, lines 55-65).

Regarding claim 30, in addition to the disclosure applied above to claim 27, the device enumeration is part of the standard USB standard discussed above. *See, e.g., Pleso* at col. 12, lines 53-66.

Regarding claim 31, *Pleso* further discloses a USB core of said host device determines a type of said peripheral device and a driver to communicate with said peripheral device (see, e.g., col. 12, lines 53-66).

Regarding claim 32, *Pleso* further discloses the determination of said peripheral device and said driver being performed in accordance with the USB specification (see, e.g., col. 12, lines 53-66).

Regarding claim 34, *Pleso* discloses: A method of installing a driver/application in a host device (see, e.g., col. 3, lines 18-31 (method for installing a peripheral device driver to a computer system)), said driver/application being required for said host device to communicate with a peripheral device (see, e.g., Abstract (a device driver which the computing system employs to communicate with the peripheral system)), said method comprising the steps of:

coupling said host device to said peripheral device (see, e.g., col. 3, lines 18-31 (connecting a peripheral device to a computer)) utilizing a USB serial interface, said peripheral device containing said software program stored in a memory device contained in said peripheral

device (see, e.g., col. 3, lines 18-31 (the peripheral device including . . . a first memory . . . storing the peripheral device driver));

[determining] if said host device comprises said driver or application in said memory device contained in said host device (see, e.g., Figure 8 (step 252); col. 11, lines 7-10 (determines whether a peripheral device driver is needed)) [and if not,] uploading said driver or application from said peripheral device to said host device over said USB serial interface (see, e.g., col. 3, lines 5-16 (the peripheral device downloads the device driver to the second memory . . . ); col. 11, lines 55-65 (USB));

installing said driver or application in said host device (see, e.g., col. 3, lines 5-16 (the peripheral device downloads the device driver to the second memory . . . )) thereby allowing communication between said host device and said peripheral device (see, e.g., Abstract (a device driver which the computing system employs to communicate with the peripheral system));

wherein said driver is associated with said peripheral device which allows for communication between said host device and said peripheral device (see, e.g., col. 3, lines 18-31 (method for installing a peripheral device driver to a computer system); col. 13, line 36, through col. 14, line 26 (the device drivers may be JAVA applications)).

Although in the context discussed above, *Pleso* fails to expressly disclose the determining and installation steps being carried out for a separate driver and application (the application being for executing a functional operation associated with the operation of the peripheral device), *Wright et al.* teaches that, in the context of providing drivers from a peripheral device through a USB interface to a host device, it has been known to similarly transfer other files (including program (application) files) relating to the peripheral device (for executing a functional operation

associated with the operation of the peripheral device) for separate installation on the host device or automatic installation during the driver installation (see col. 5, lines 63-67). Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate such additional application determining and installing as per the teachings of Wright in order to gain the advantage of combining related installations for a peripheral device.

Regarding claim 35, in addition to the disclosure and teachings applied above to claim 25, although *Pleso* fails to expressly disclose the sequential determining regarding the application and driver, *Wright et al.* teaches that, in the context of providing drivers from a peripheral device through a USB interface to a host device, it has been known to similarly transfer other files (including program (application) files) relating to the peripheral device for separate installation on the host device or automatic installation during the driver installation (see col. 5, lines 63-67). Therefore, it would have been obvious to one of ordinary skill in the art to incorporate such sequential determining steps as per the driver and application installation system taught by Wright in order to gain the advantage of combining related installations for a peripheral device.

Regarding claim 36, *Pleso* further discloses said host device and said peripheral device communicate with one another utilizing the USB specification (see, e.g., col. 11, lines 55-65).

Regarding claim 37, in addition to the disclosure applied above to claim 34, the device enumeration is part of the standard USB standard discussed above. *See, e.g., Pleso* at col. 12, lines 53-66.

Regarding claim 38, *Pleso* further discloses a USB core of said host device determines a type of said peripheral device and a driver to communicate with said peripheral device (see, e.g., col. 12, lines 53-66).

Regarding claim 39, *Pleso* further discloses the determination of said peripheral device and said driver being performed in accordance with the USB specification (see, e.g., col. 12, lines 53-66).

8. Claims 33 and 40 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,009,480 (Pleso), Srinivas Yarra, "USB OTG software frees dual-role handheld devices," May 16, 2002, EDN, pp. 83, 84, 86, and 88 (herinafter "Yarra"), and U.S. Patent No. 6,754,725 (Wright et al.).

Regarding claims 33 and 40, in addition to the disclosure and teachings applied above to claims 27 and 34, although *Pleso* discloses the use of a USB interface for host-peripheral communications, *Pleso* is silent on the use of the USB On-The-Go (OTG) specification to define roles of host devices and peripheral devices. However, *Yarra* teaches the OTG specification as being a supplement to the USB standard and adding several unique advantages over the previous standard, including the determination of host and peripheral roles (first through a cable select mechanism and additionally through software negotiation of roles). *See, e.g., Yarra* at pp. 83-84 (The A device is the default host and the B device is the default peripheral. Through the Host Negotiation Protocol, the host functions can be transferred to the B device). Therefore, because the USB OTG specification, (1) was a known supplement to the standard USB specification, (2) provides known tangible benefits over the USB specification, and (3) is being used for its

intended purpose (i.e., determining the role of host and peripheral in serial communications), it would have been obvious to one of ordinary skill in the art at the time the invention was made to utilize such a USB OTG specification with the system of *Pleso*.

### ***Conclusion***

9. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

10. Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Eric B. Kiss whose telephone number is (571) 272-3699. The Examiner can normally be reached on Tue. - Fri., 7:00 am - 4:30 pm. The Examiner can also be reached on alternate Mondays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Tuan Dam, can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Eric B. Kiss/  
Eric B. Kiss  
Primary Examiner, Art Unit 2192